Chris Tully's HCAL Pulse Shape in ORCA 3_0_5

All real work was done by Chris Tully/ Teresa Monteiro!
I just look at it/test it. the HCAL group is very very lucky that these two ecal people have kindly done this coding for us.

Sarah Eno

Offline

Basic Flow (in Fortran-ese)

"timesamples" exists for each calorimeter cell, and are stored in a set of objects called CaloPileUp's (one object for each detector). The timesample is basically an array, and for ECAL and HCAL has a length of 10 samples (25 ns sampling interval)

Hits from GEANT for pileup are added to the "timesamples" by CaloCommon/CaloPileUp.cc *(code duplicates the EcalRUFrom… code)* when the CaloPileup is created

Hits from GEANT for the hard-scattering are added to the CaloPileUp corresponding to the cell the hit occurred in by G3EcalDigits/EcalRUFromReadoutSimulation.cc *(don't be fooled by name. Does this for hcal, vcal, and preshower as well)*

noise is injected by EcalFrontEndSimulation.cc, maybe thru a "output filter" mechanism. other filters can do selective readouts and trigger primitive generation from the timesamples.

timesamples are converted to energies by G3EcalDigits/EcalDigitsFromReadoutSimulation.cc

EcalRUFromReadoutSimulation.cc

What is does

1) loop over the OO version of the GEANT hits for the hard scattering. The OO version contains information about the cell that the hit occurred in. This allows you to find the timesample for that cell in the pileup list. The time structure of the hit is calculated, and it is added bin-by-bin to the timesample.

2) after this is done, the result should be converted to integers to represent the ADC granularity. this will be done in ORCA 4

EcalRUFromReadoutSimulation has a private member, pileup, that points to CaloPileUp's, made by CaloPileUp. The CaloPileUp objects contain the timesamples, in a private member called tsbuffer.

HcalChannelSetup.cc

in Calorimetry/G3HcalDigits/src/

Contains (amoung other routines)

HcalVShape::computeShape() - used to initialize the shape for HCAL pulse

HcalVShape::value() - used to get shape at a time (used in converting a GEANT hit into a timesample)

shape

From Dan Green….

// unit height in GeV and time constants in ns
const float a = 0.0269257; // pulse height normalization for 1 GeV of energy
const int ts = 11; // scintillation time constant
const int thpd = 10; // HPD current collection drift time
const int tpre = 25; // preamp time constant (should this be 3ns? Tully was unsure)

$$f(t) = \frac{C\int_0^t n_{td}(t_d)\int_0^t n_{th}(t_h - t_d)\,n_{tp}(t - t_h)\,dt_h\,dt_d}{\int_0^\infty n_{td}(t_d)\,dt_d \int_0^\infty n_{th}(t_h)\,dt_h \int_0^\infty n_{tp}(t_p)\,dt_p}$$

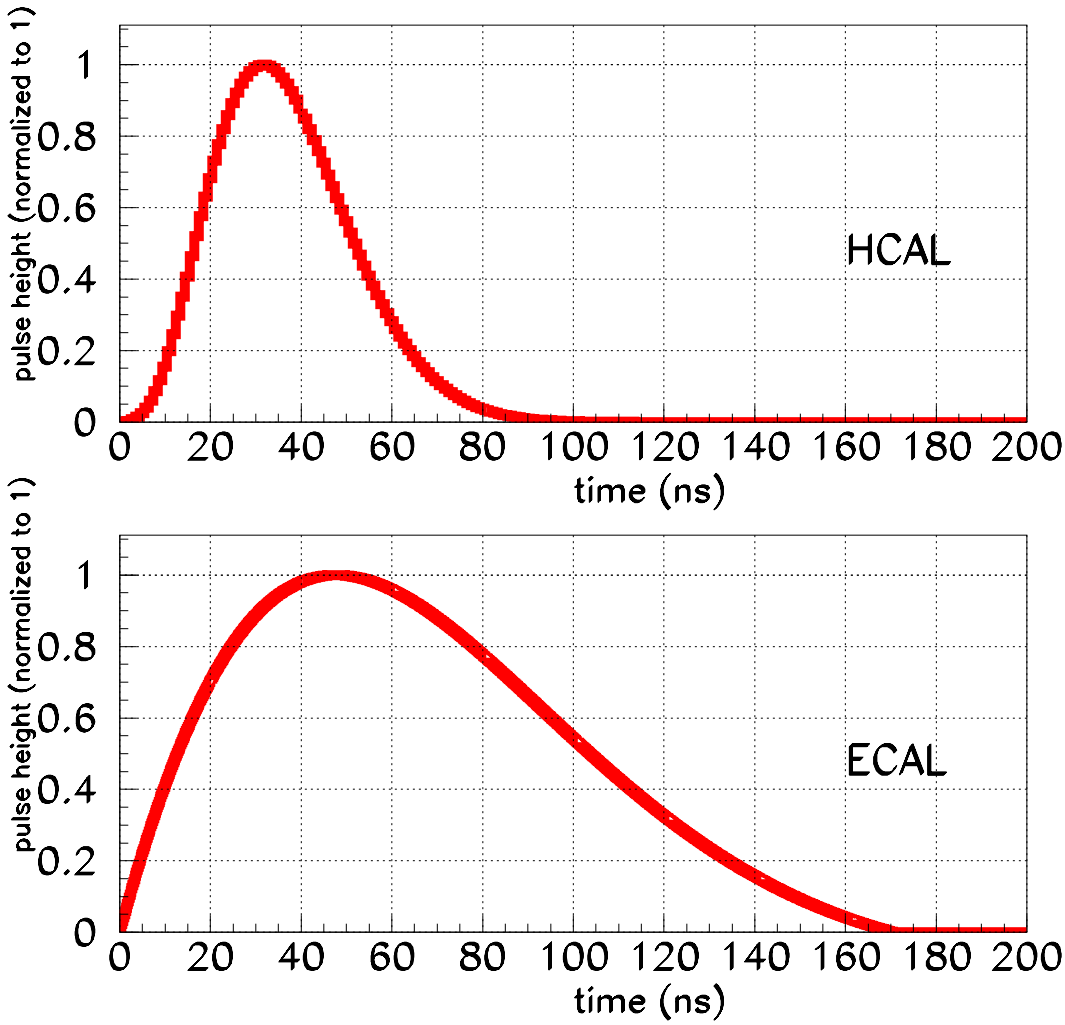
$$n_{td}(t) = \exp(-t/\tau_s)$$

$$n_{th}(t) = 1.0 + (t/\tau_{hpd})$$

$$n_{tp}(t) = t \bullet \exp(-(t/\tau_{pre})^2)$$

Used both in HCAL and VCAL!! Is this okay?

shape


pulse height (normalized to 1)
1
0.8
0.6
0.4
0.2
0
0 20 40 60 80 100 120 140 160 180 200
time (ns)
HCAL
pulse height (normalized to 1)
1
0.8
0.6
0.4
0.2
0
0 20 40 60 80 100 120 140 160 180 200
time (ns)
ECAL

How shape fills timesamples

Timesamples are basically arrays with size of 10

amplitude for each time sample is calculated as

amp = shape (bintime) * geant_energy

bintime = 25ns (bin number-5) + 32.0 - jitter + crossing_offset (hcal)

=-93,-68,-43,-18,7,32,57,82,107,132 ns

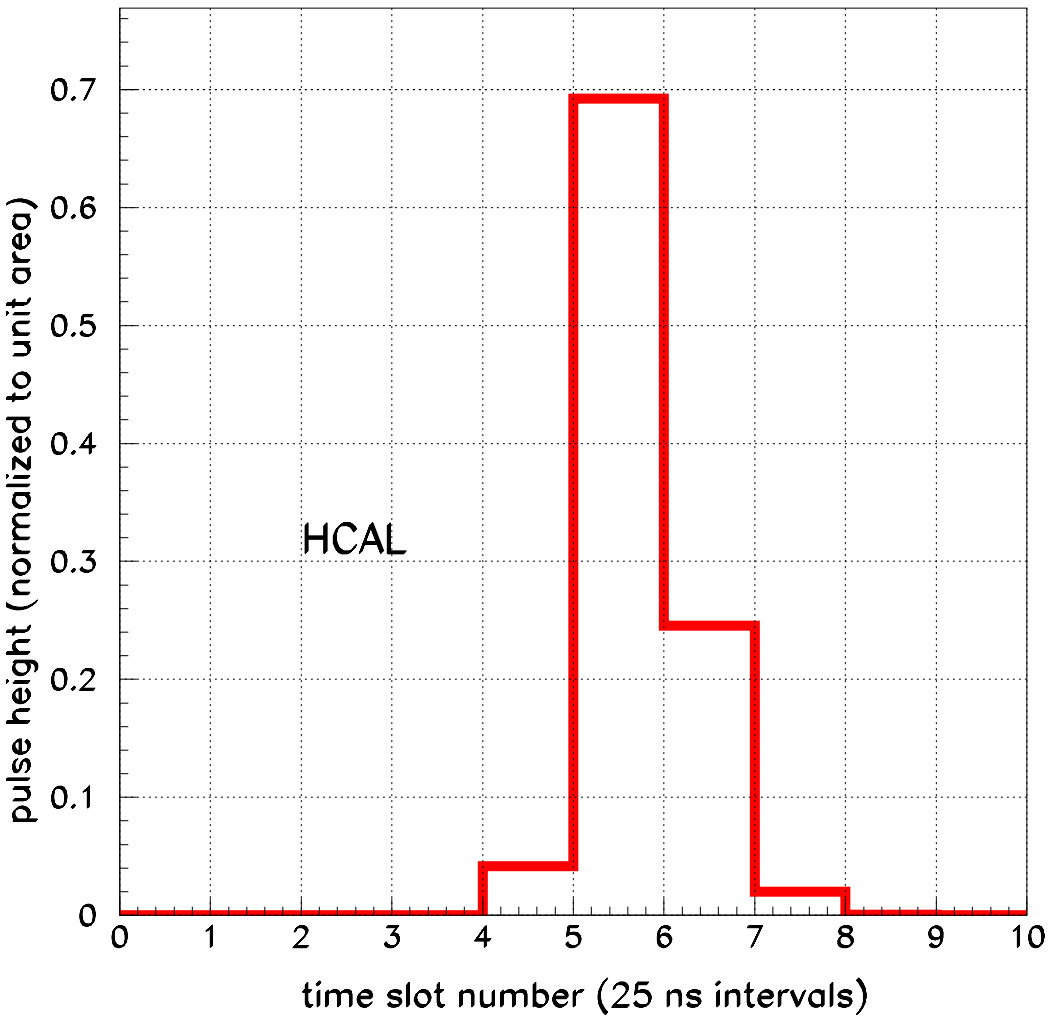
=25 ns (bin number-5) + 47.6683 -jitter + crossing_offset (ecal)

crossing_offset is for energy from previous/later crossings (occurs in 25 ns steps). =0 for hard scatter.

jitter should be geant_hit_time - estimated_hit_time, but for now is set to zero for hcal.

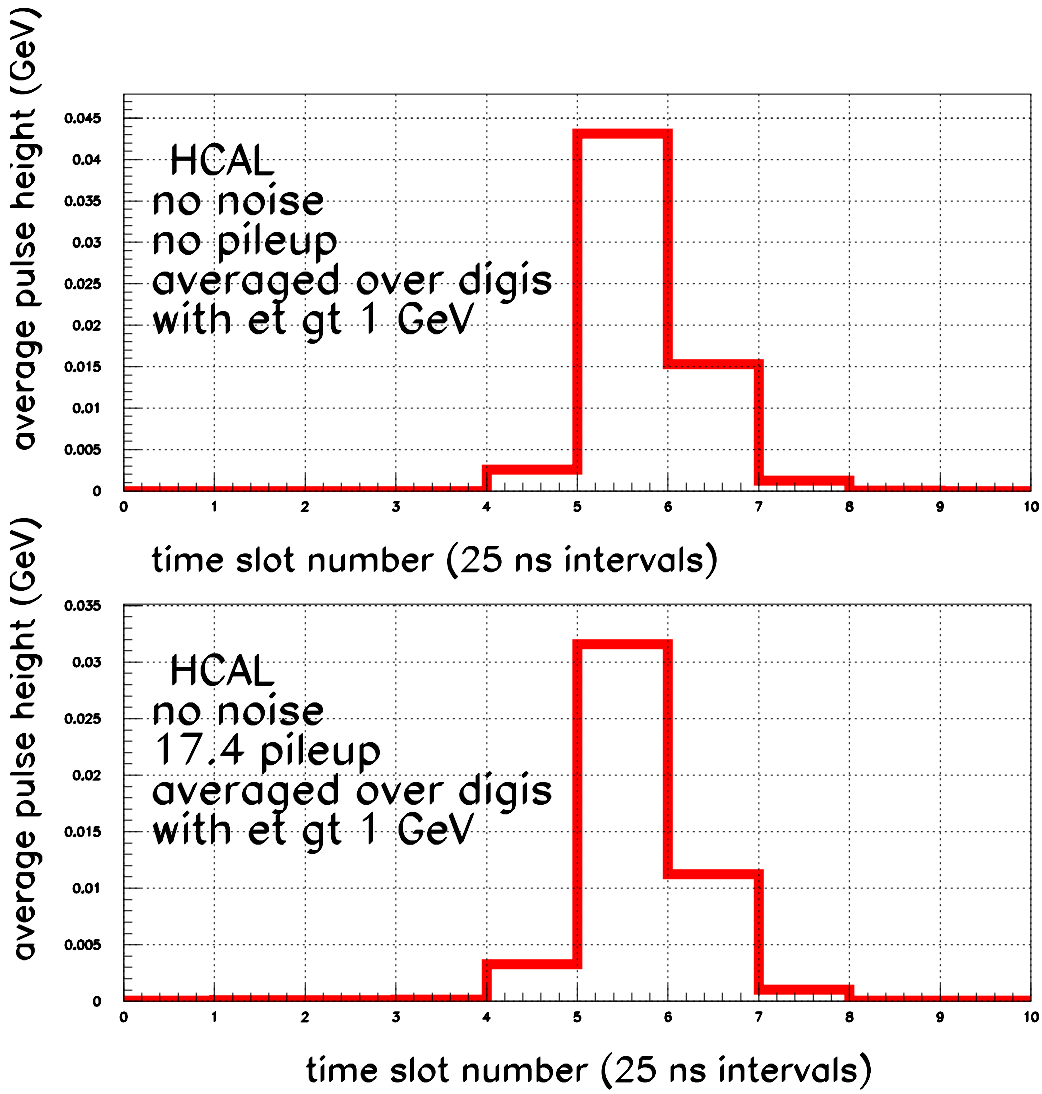
Same for HCAL and VCAL!

Example: single 30 GeV pions, eta=0.4, no noise, no pileup

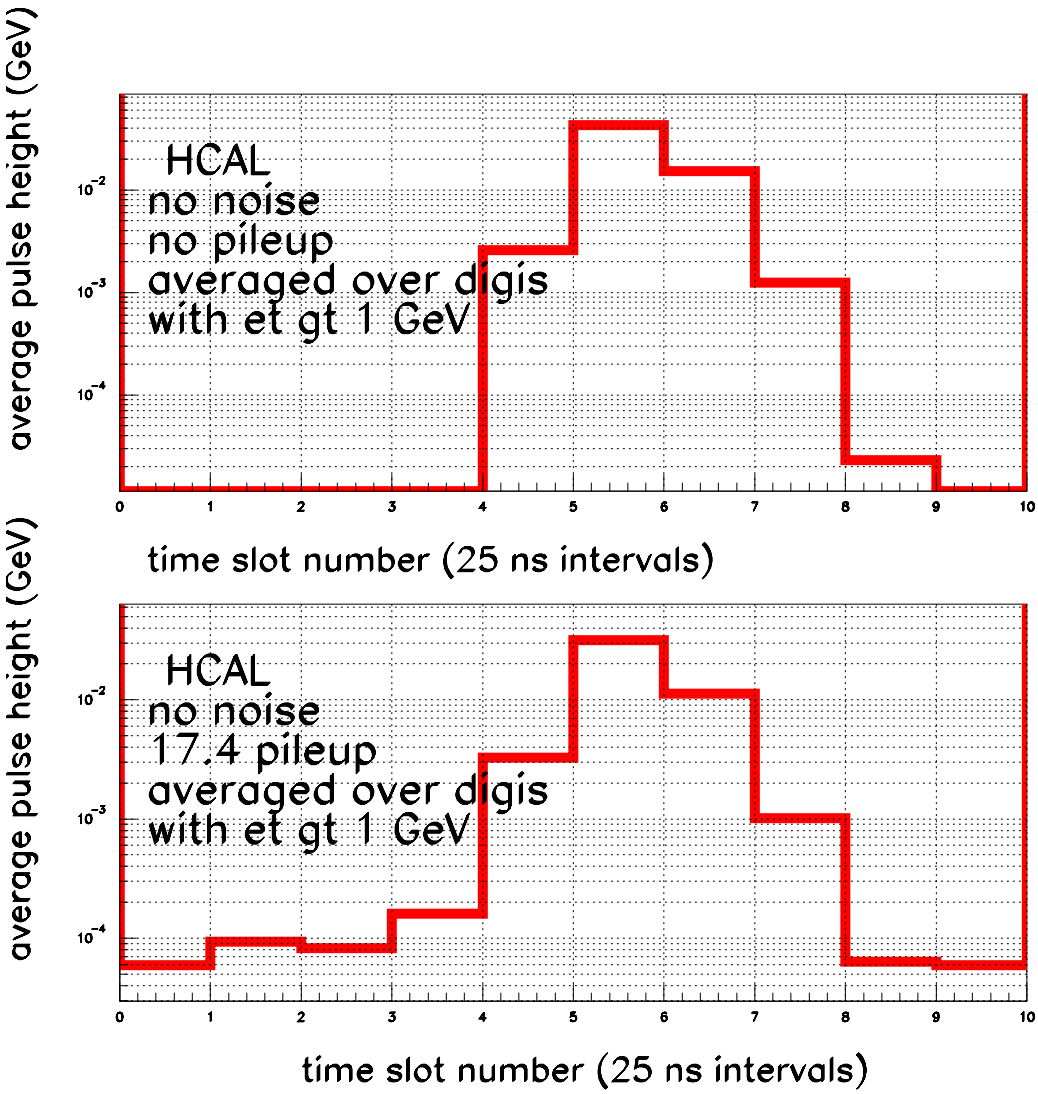


10 time samples
Code arranges it so max pulse height should occur in bin 5

With 17.4 interactions



With 17.4 interactions



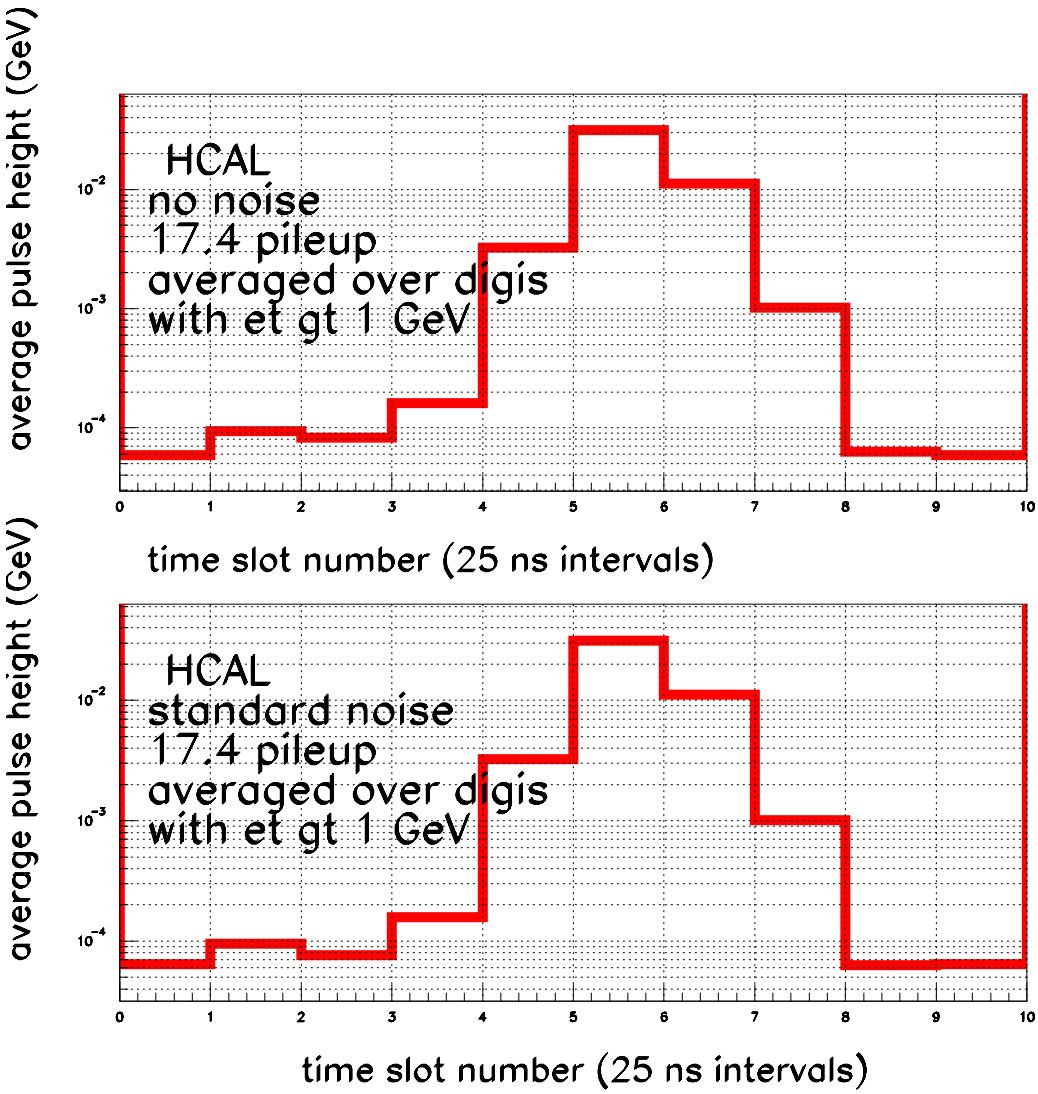
With noise?

Right now, noise is set at 0.0001xsampling correction

- HB1: 7.2 MeV
- HB2: 14.7 MeV
- HB3: 14.7 MeV
- HB4: 20.0 MeV
- HE1: 10.8 MeV
- HE2: 23.7 MeV
- HE3: 23.7 MeV
- VCAL1: 0.207 MeV
- VCAL2: 0.140 MeV
- VCAL3: 0.0

Have no idea if this is right, but this (as you'll see later) is a very important part of any study about HCAL pulse shape

With noise?


HCAL
no noise
17.4 pileup
averaged over digis
with et gt 1 GeV
average pulse height (GeV)
time slot number (25 ns intervals)
HCAL
standard noise
17.4 pileup
averaged over digis
with et gt 1 GeV
average pulse height (GeV)
time slot number (25 ns intervals)

EcalDigitsFromReadout Simulation.cc

What it does:

1) dig the timescales out of the complicated storage structure (CaloRU lists contain CaloRU's, which contain CaloTower's, which contain CaloTowerCells. From the CaloTowerCells, we can get properties like shapes associated with that cell, and use information in this to search the CaloPileUp objects for the timesample associated with this cell.

2) using that pileup, and the weights, sampling scale factors and other stuff, which are stored in objects that are getable from the CaloTowerCell, calculate the energy, jitter, and chi2 for that timesample. Especially, it uses the routine evalAmplitude from G3HcalDigits/src/HcalChannelSetup to do this.

HcalChannelSetup.cc

in Calorimetry/G3HcalDigits/src/

Contains (amoung other routines)

HcalVAnalyser() - used to set the depth scale factors

evalAmplitude() - takes a timesample and uses it to calculate the energy.

How the energy is calculated

Basicially energy = sum(w_I * timesample(I))

	Weights	
bin number	HCAL/VCAL	ECAL
0	0	0
1	-0.208	-0.361
2	-0.208	-0.361
3	-0.208	-0.361
4	-0.138	0.271
5	0.942	0.467
6	0.201	0.347
7	-0.174	0.111
8	-0.207	-0.112
9	0	0

Hcal calculation basically uses bins 5 and 6
negative weights do the pedistal subtraction

How the Energy is Calculated

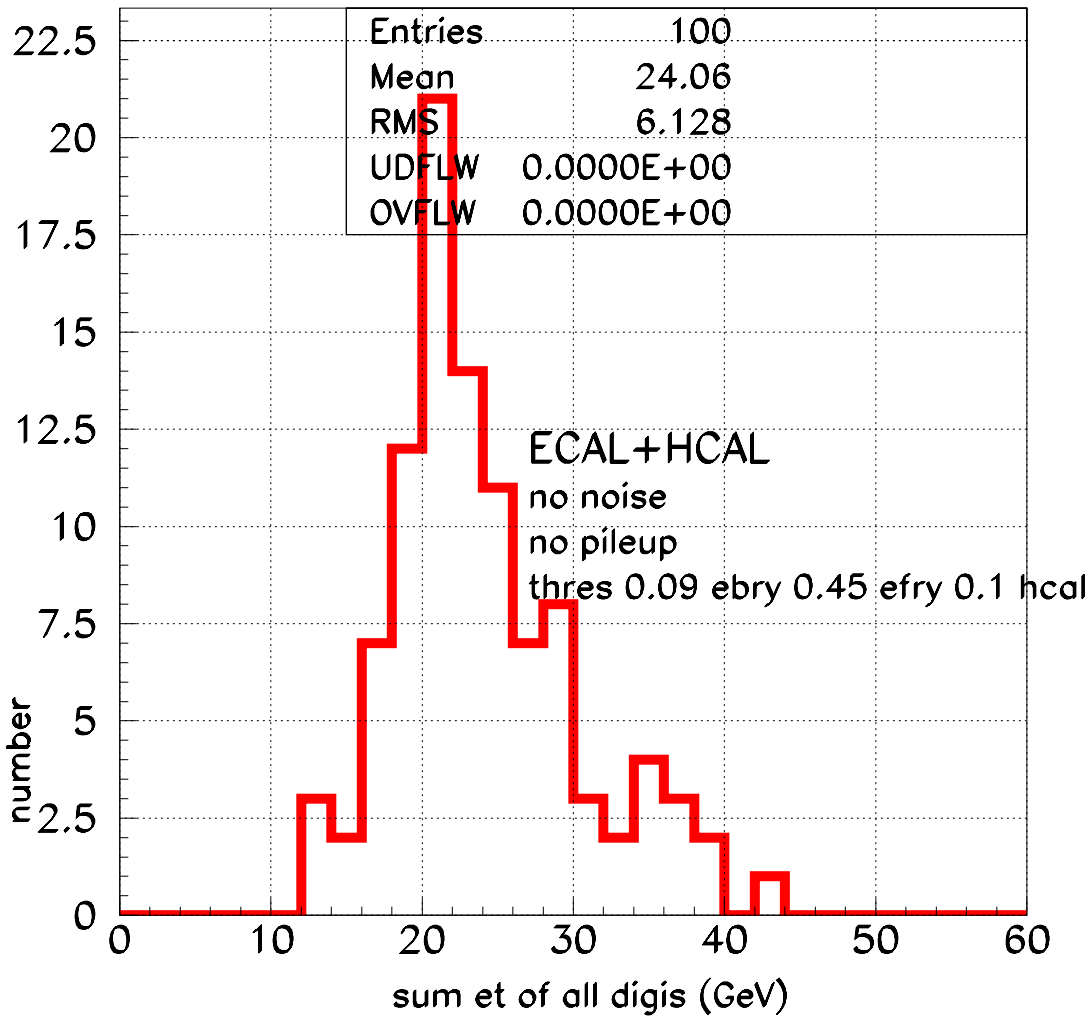
Results are scaled by the sampling correction factors

```
// From Shuichi Kunori
//HB  depth-1     72.
//          2    147.
//          3    147.
//          4    200.
//HE  depth 1    108.
//          2    237.
//          3    237.
//HF  depth 1    2.07
//          2    1.40
//          3    0.00
//   (Note- Ring variation in HF is ignored.)
```

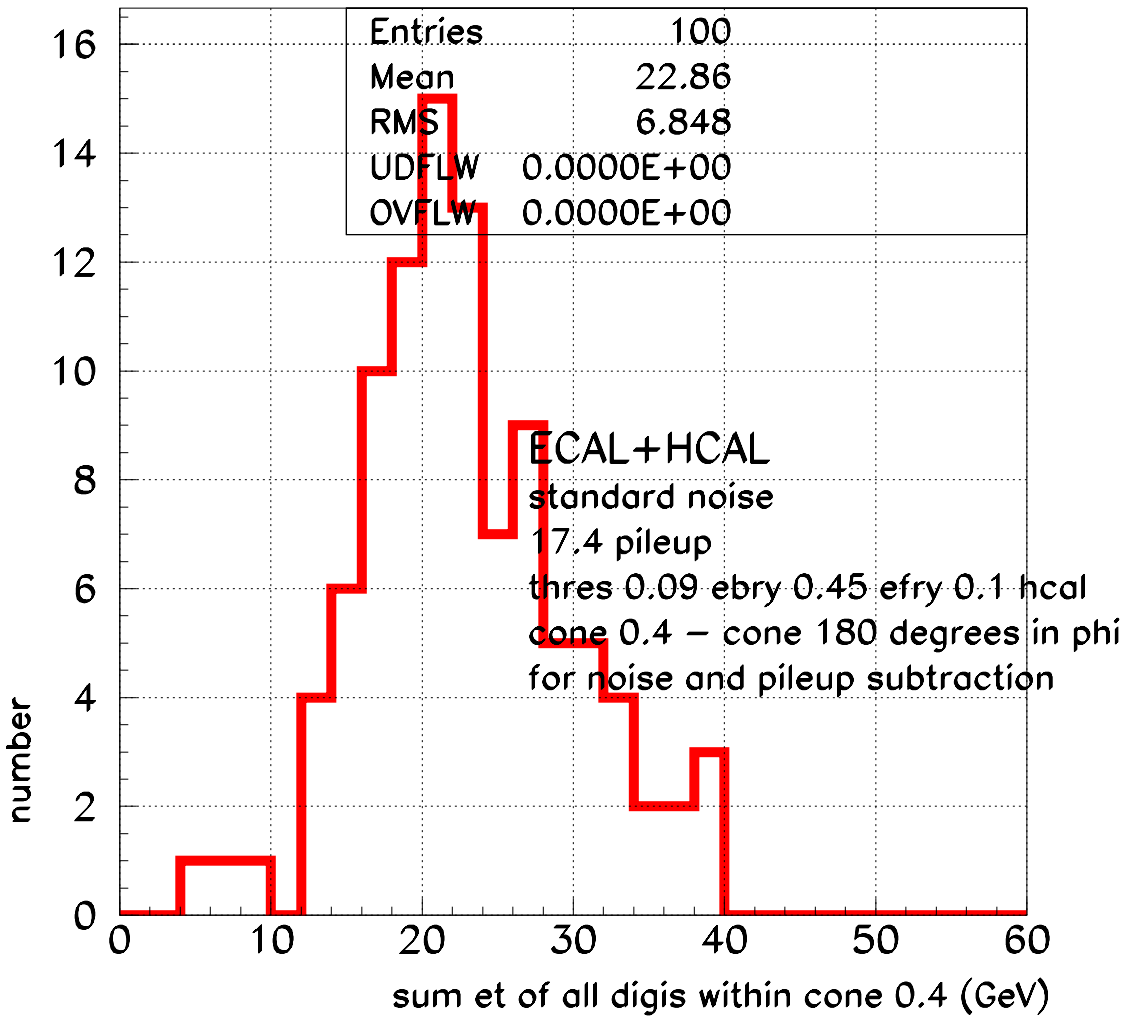
note, HB depths 2+3 are actually ganged in hardware. they are not separate readouts. same for HE depths 2 and 3.

(note, Shuichi says radius is wrong in HcalBase.cc is 180,190,300,320, should be 180,190,287,406). how to arrange for this change?

Energy for single 30 GeV Pions, eta=0.4, no noise, no pile



With pileup

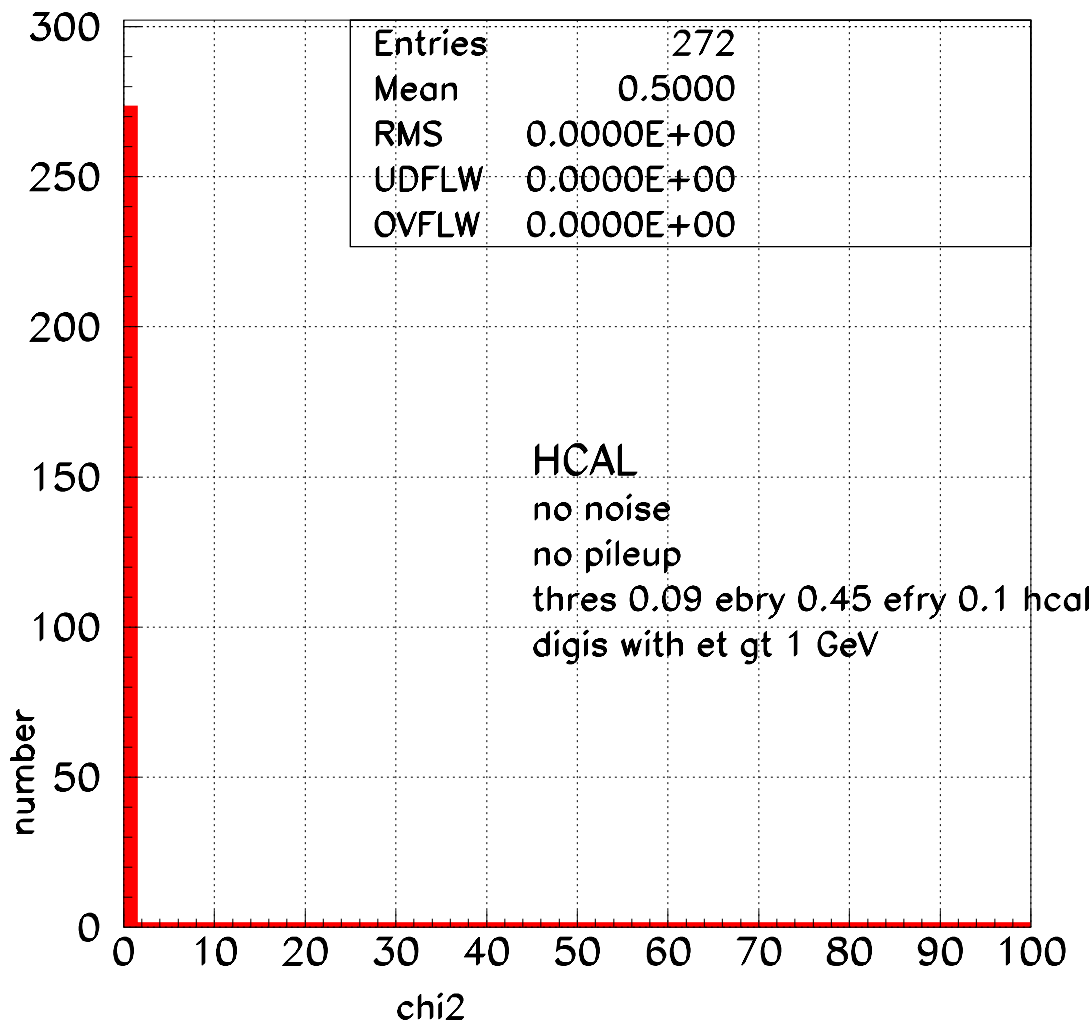


Cone(0.4)- cone(0.4) 180 degrees in phi away

Chi^2

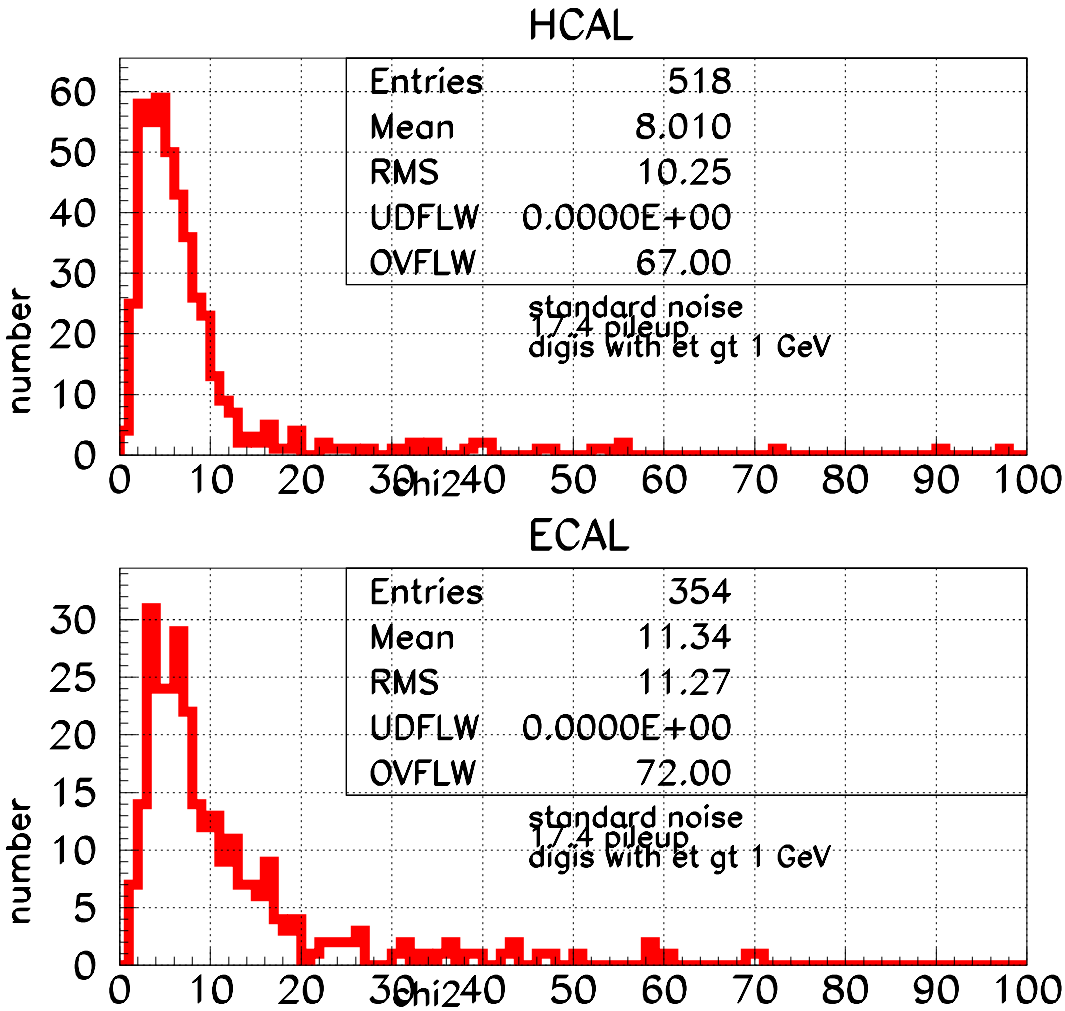
It also calculates a chi^2 to see if the timesample shape is consistent with the assumed HCAL shape

Chi^2 dist for single pions, no noise, no pileup

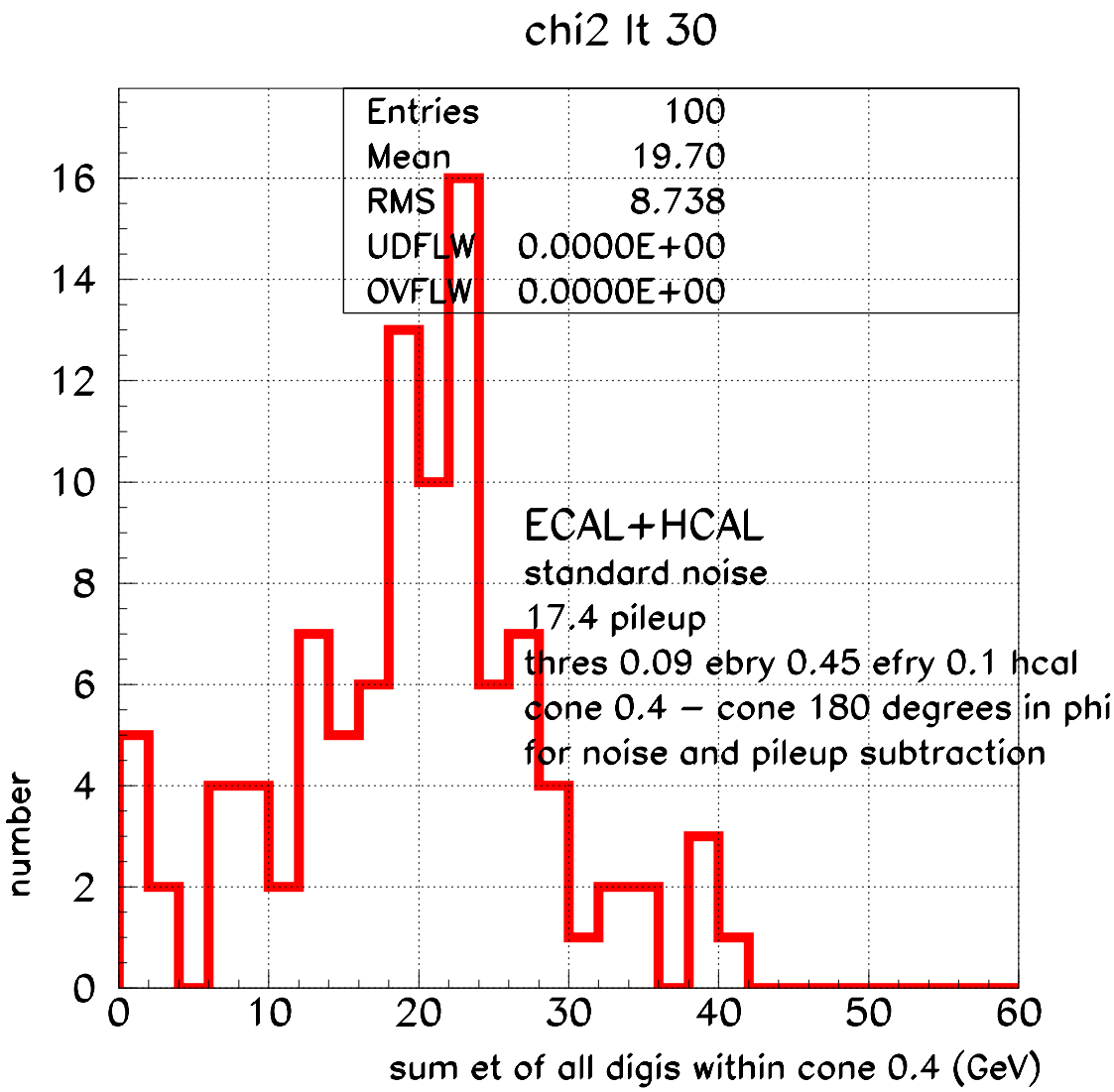


Always zero of course!

With 17.4 interactions + noise



Et with chi2< 30 cut



warning

Chi^2 test is not applied to digis before filling ecalplushcaltowers

should it?

Zeroing jitter

In CaloCommon/CaloHitsFromCMSIM.cc

```
//  const double tcalbr=1.0E-09;
const double tcalbr = 1.0;
```

(used to calculate time of hit from GEANT info)

wrong for HCAL for CMSIM prior to CMS118 (should be 10.0)
caltech production used CMS116

got around this by lines like

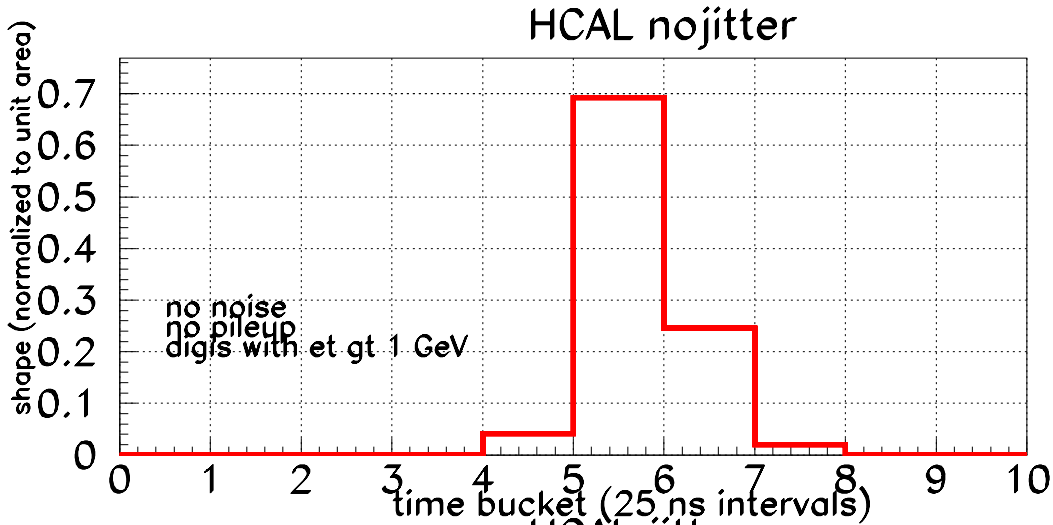
```
double jitter = 0.;
                    if( (id.WhichDetector() == "HCAL")
                     || (id.WhichDetector() == "ESFX") ) {
                      jitter = timeOfBunch;
                    } else {
                      jitter = thit -id.TimeOfFlight();
                    }
```

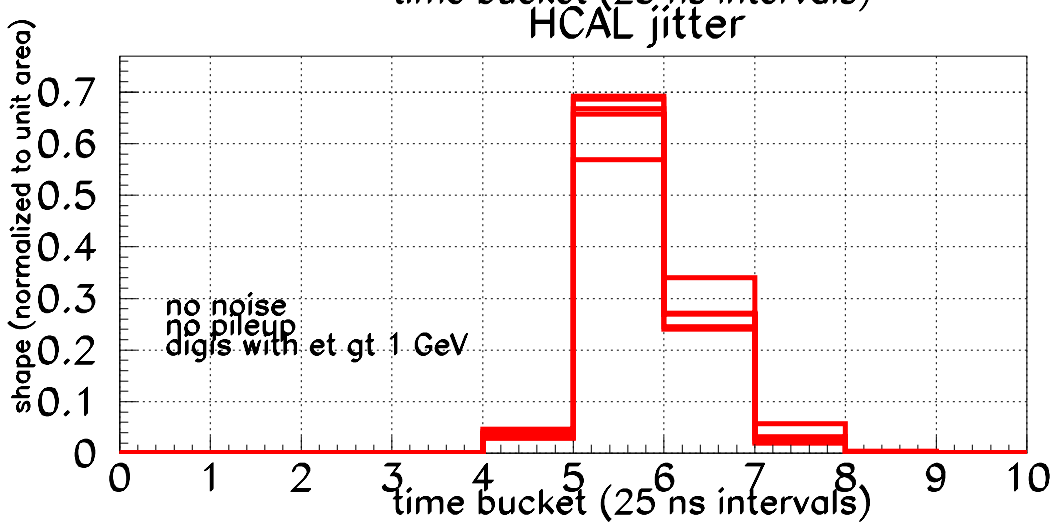
in CaloPileUp.cc and EcalRUFromReadoutSimulation.cc

Zeroing Jitter

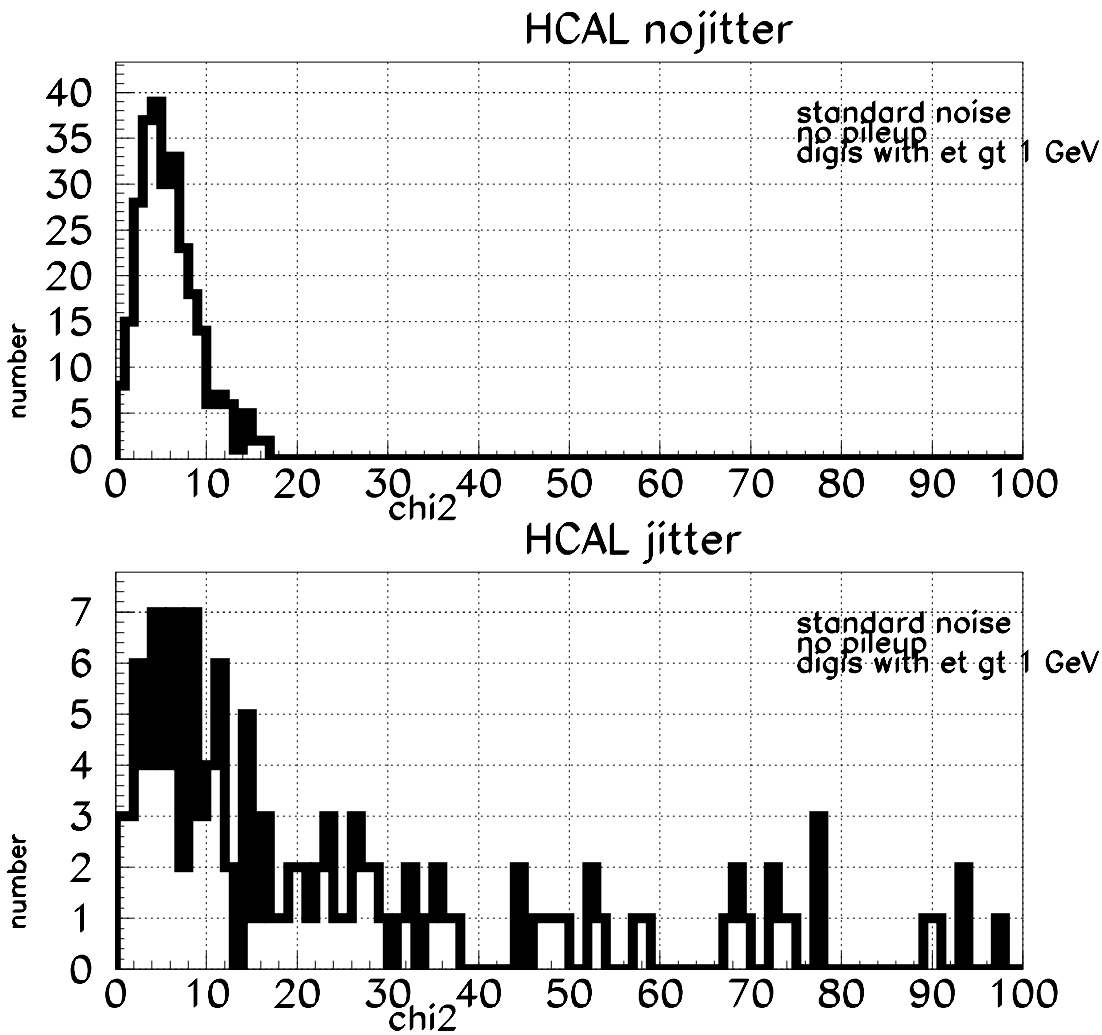
Set hcal.tz so that GEANT uses 1 ns, not 10 ns and make some 30 GeV pions.

Time dist for 5 separate digis





Affect on Chi2



L1

How do you decide if a cell contributes to the “current” bunch crossing?

Calorimetry/G3EcalDigits/src/CaloTriggerTowerFilter.cc

Need something similar to chi^2

L1: ECAL

combine the timesamples from all cells corresponding to a tower

Calculate the energy using the weights, assuming the current bunch crossing, and assuming +-1 bunch crossings.

If it is maximum in the current bunch crossing, it is used…

method Jitter() in class EvalAmplitude
Calorimetry/CaloCommon/interface/EvalAmplitude.h
returns 0 if max

L1: Hcal

Loop over 4 layers

for each layer, apply weights for the 10 time slices, extract energy, multiply by sampling correction factor (no test is done to see if the distribution in time is consistent with the "current" bunch crossing)

sum the result

should we sum, then extract, or extract and then sum? With what weights? 10 time slices? What sampling correction factors? Should we do time-distribution consistent test (code exists, is just not used)

L1

Energy is extracted from the combined timesample just as in the offline? (using the weights, and the same weights)

```
// From Shuichi Kunori
//HB  depth-1     72.
//          2    147.
//          3    147.
//          4    200.
//HE  depth 1    108.
//          2    237.
//          3    237.
//HF  depth 1    2.07
//          2    1.40
//          3    0.00
//   (Note- Ring variation in HF is ignored.)
```

I understand that layers 2+3 are ganged in hardware. L1 should only use 1,2, and 3 (not 4). We should change this in the code? (now it uses 4). I understand that the two that are used are summed *before the energy extraction? With what weights?

Other issues

Are there 2 hardware bunch crossing determinations, one for L1, and one for L2? What is the purpose of the L2 one? What is it used for? How does the algorithm differ from the L1 one?

ORCA 4

Note: my understanding is that all this structure will change in ORCA 4 by David Chamont's group

Conclusion

To do list

1) how to use chi2
2) what to code for L1
    1) get rid of 4th layer
    2) combine then extract (with what weights) or extract then combine?
    3) should an equivalent of Jitter be turned on for hcal?